Abstract Presentation Back to Agenda Next Abstract

VSIPL for Diverse Architectures (Pentium 4 to DSPs)

Mr. Brian Chase
Mr. WenhaoWu
Dr. Anthony Skjellum
MPI Software Technology, Inc
Phone: (662) 320-4300, ext. 13
Fax: (662) 320-4301
E-mail: brian@mpi-softtech.com
E-mail: wenhao@mpi-softtech.com
E-mail: tony@mpi-softtech.com

Many companies have described their experiences with using Motorola G4 processors to provide the VSIPL CoreLite standard for military and medical computing, and others have described their efforts with highly optimized Core and Core+ profiles.

This poster uses the experience of MPI Software Technology's existing Core+ optimized VSIPL implementation for G4 as the springboard for supporting other platforms of emerging interest to COTS, defense, medical and imaging customers.

Three kinds of technologies are occurring in military computing that may unseat the preeminence of low-power G4-style RISC processors: 1) The growing power of DSPs including better development environments, as typified by TI TMS320C6x family (as opposed to SHARC 2106x, which was harder to program strictly from C and libraries), 2) The use of medium-power systems with Pentium 4 based blades, 3) The growing perception that PowerPC is lagging Pentium in overall performance

Furthermore, the clock speed of Intel Pentium line of processors is now reaching three or more GHz. Also, the vector processing registers (SSE) available on the Pentium III or later provide several orders of improved performance for single precision floating point operations. Additionally, the well-known Pentium family provides a cost-effective COTS solution for embedded hardware designers as well as the end users. This means that although high power consumption, their Flops/Watt are becoming more attractive.

MPI Software Technology now has a fully optimized VSIPL (Core and CoreLite) library, known as VSI/Pro, for Pentium 4 / SSE platform targeting Linux, Windows, and VxWorks operating systems. The FFT performance of VSI/Pro is consistently better than Intel’s Math Kernel Library (a.k.a MKL), as illustrated in the following graph. VSI/Pro has a carefully designed API, i.e., VSIPL, which is well fitted to the needs of the signal and image processing community in the defense and medical industries.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **20 AUG 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **VSIPL for Diverse Architectures (Pentium 4 to DSPs)**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **MPI Software Technology, Inc**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

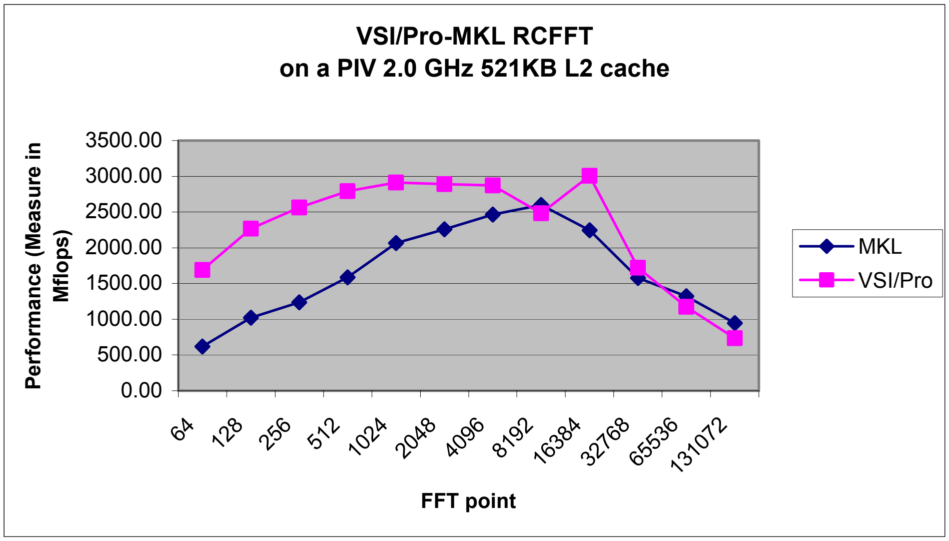
13. SUPPLEMENTARY NOTES
See also ADM001694, HPEC-6-Vol 1 ESC-TR-2003-081; High Performance Embedded Computing (HPEC) Workshop (7th)., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**22**	

Abstract
Presentation
Back to Agenda
Next Abstract



The Texas Instrument's TMS320C67 family of processors is a general purpose DSP chip that is specifically designed for FFT and FIR operations. This family of processor is widely used and accepted by the signal processing community with the applications including software radio, modems, and sonar. Unique features of this family of processor include: 1) a very deep pipeline and 2) a very large instruction word (VLIW) architecture. Texas Instruments provides an integrated development environment known as Code Composer, which supports the C and C++ programming languages. Also, Tiny C Compiler has the ability to produce correct code for this DSP chip. The porting of VSI/Pro to this family of processor has started by the hand tuning of FFT operations and the algorithm mapping from the original RISC/CISC implementation to DSP chip.

Experiences, results, plans and future work for supporting Intel Pentium 4 and Texas Instruments TMS320C67 family within VSI/Pro are described in this poster.

Abstract Presentation

Back to Agenda
Next Abstract

MPI Software Technology, Inc.

VSIPL for Diverse Architectures (Pentium 4 to DSPs)

Anthony Skjellum
Brian Chase
Wenhao Wu

September 23, 2003

WELCOME
TO OUR WORLD.

Prelude – VSI/Pro

- Overview: Current VSIPL platform support
- Status: G4 / Altivec
- Widely used worldwide
- Domestic production computing adoption picking up
- Helps untie programs from specific vendors
- Expertise on optimizing G4 a major part of expertise
- Expertise on porting to different PPC environments also key expertise
- Dealing with C/C++ toolchains a major expertise
- Key optimizations for more advanced users (e.g., Rader's algorithm and other NTT-motivated improved) with high performance are at cusp of newest release efforts
- Complete version for Image processing also released
- Customers have started asking for non-G4/Altivec alternatives!

MSTI’s Strategy

Availability on Different Processors	Operating System / Development Tools
• Core+ G4 / Altivec	• VxWorks, MercuryOS, LynxOS, Linux, MacOSX
• Core P4 / SSE	• Windows, Linux, VxWorks
• CoreLite TI DSP C67 family	• Code Composer toolset

Why P4?

- The higher clock speed, 3 or more GHz
- COTS technology enables cost effective solutions
- Anticipated lower power versions from Intel and third parties in future
- Not all embedded systems equally power/heat constrained even now
- Double precision 4-way vectorization useful
- Future winner in Gflop/Watt? Gflop/$?

Why C67

- Specially designed architecture for DSP applications.
  - very deep pipeline
  - very large instruction word (VLIW) architecture
  - streaming data
- Better GFlops per $ than G4 / Altivec

Product Exploration/Results - P4

- Full Core profile support for Windows, Linux, and VxWorks.
- Optimized FFT performance for SSE registers (performance graph later)
- Optimized matrix library easily achievable also
- Can equal or beat MKL (Intel commercial library) in significant aspects of overall performance… more tuning possible

Porting Experience for C67

- What we achieved in 1 month:
- VSI/Pro Core Lite profile is completely ported for TI C67.
- We have C6711 optimized Complex-to-Complex inplace and out of place, forward and inverse FFTs:
  - vsip_ccfftop_f()
  - vsip_ccfftip_f()
  - C6711 150Mhz CPU 29300 cycles for 1024 element FFT

Issues

- C side: Straightforward
- C++ side: Strict on template support
- VLIW assembly side: No hand tuned assembly code in the library yet… next step before product release

C67 Operating Systems

Example Various OS platforms:

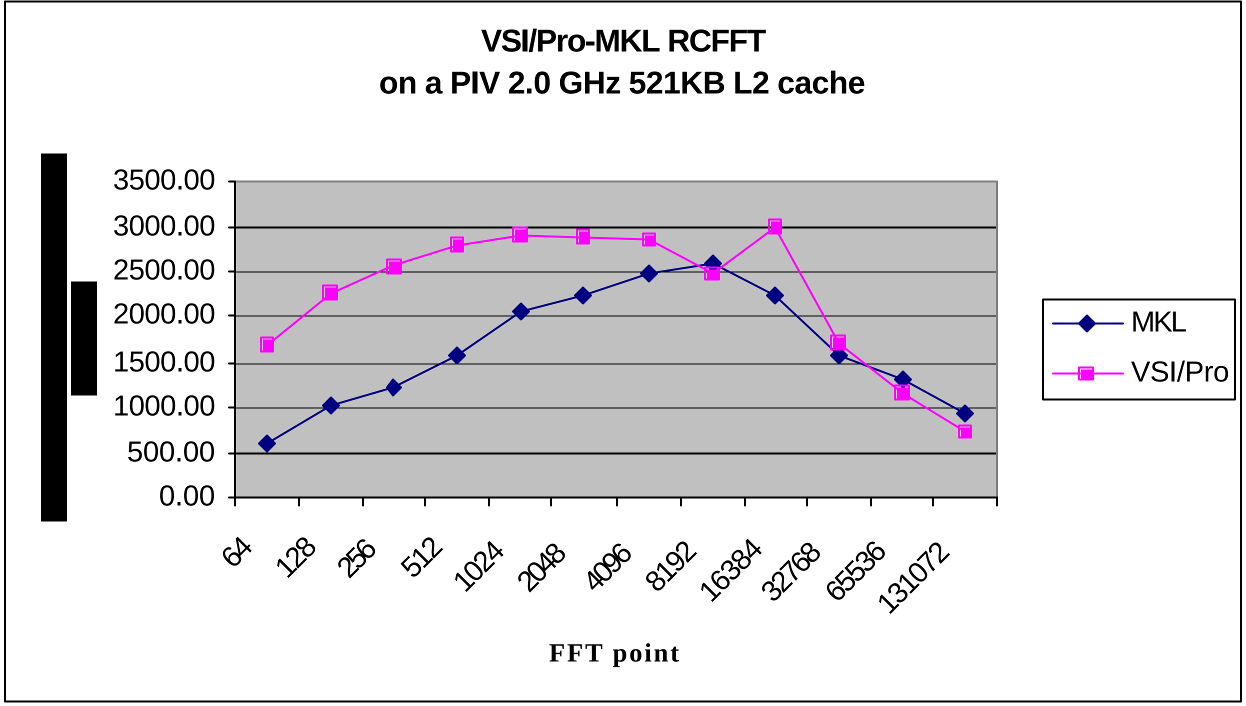
- SPARK (Small Portable Adjustable Real-time Kernel)
- OSE
- Diamond
- Thread XABS GmbH Jena

FFT optimization for P4

- Algorithm was engineered for the architecture to minimize problems arising from the scarcity of registers and lower cache associativity.
- The algorithm is auto-sort DIF, efficient not only on power-of-two sizes.
- The key functions are written in assembler supported by highly optimized C and C++ code, using SSE.

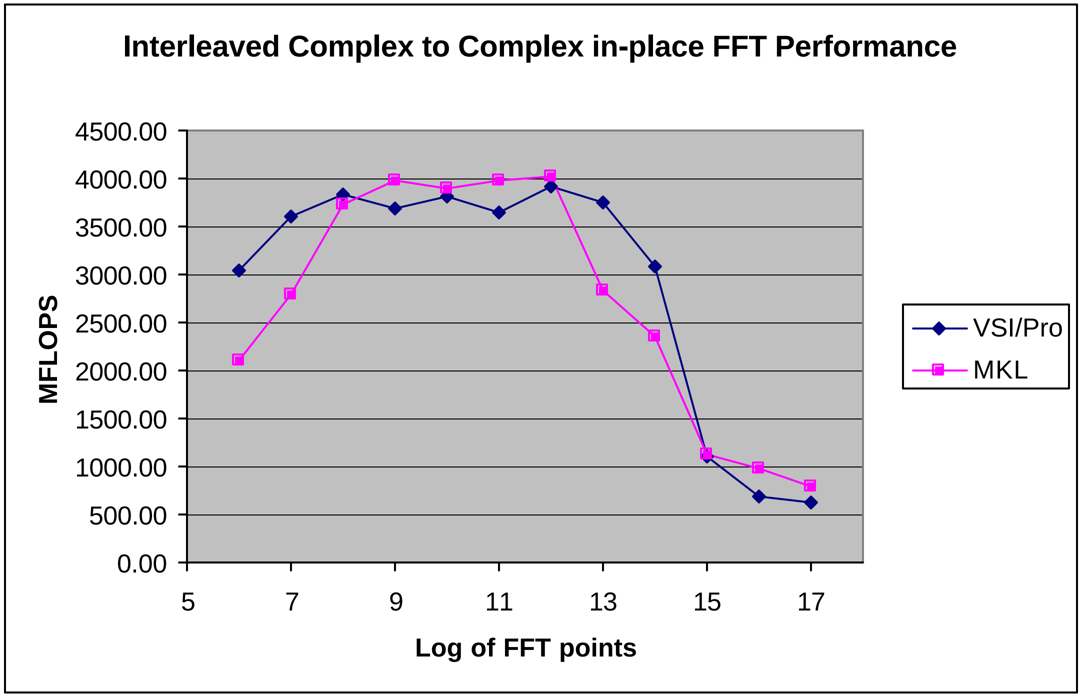
FFT performance on P4

- RCFFT Comparison with MKL…



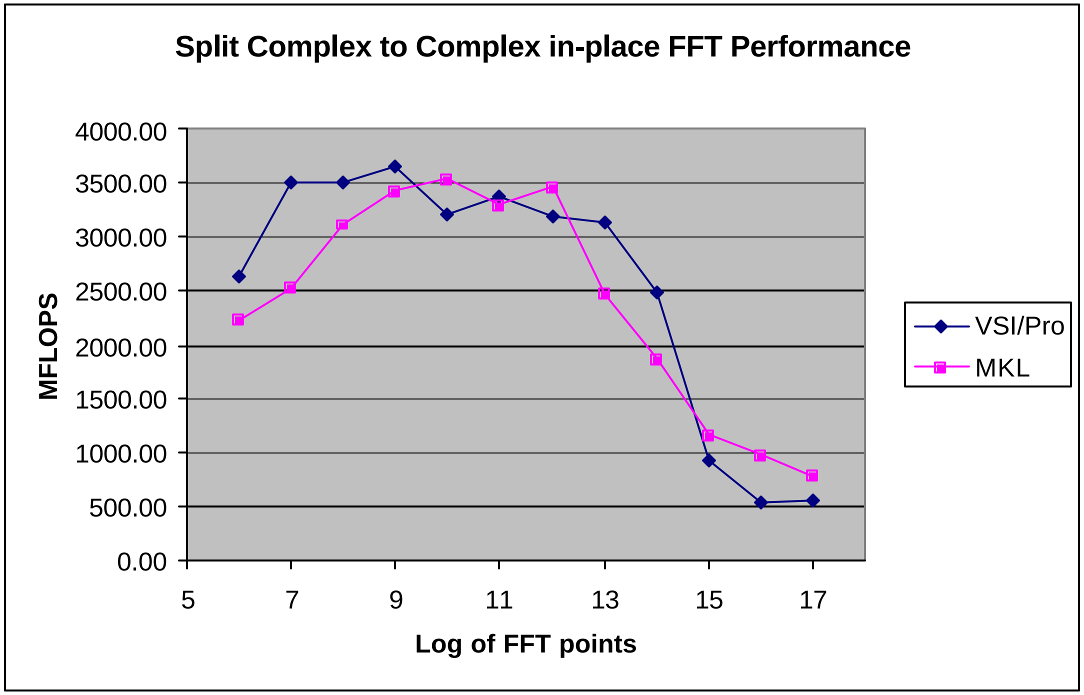
FFT performance on P4

Interleaved in-place CCFFT Comparison



FFT performance on P4

Split in-place CCFFT



FFT optimization for DSP

- Using Radix-2 , Radix-4 algorithms.
- Also using cache splitting (the L1 cache is 4KB, so the splitting is needed for sizes > 256 for in-place FFT) .

PRELIMINARY
FFT performance on C67

TI DSP C6711 150 MHz DSK

	ccfftop fwd	ccfftop inv	ccfftip fwd	ccfftip inv
Length(N)	scale = 1.0	scale = 1/N	scale = 1.0	scale = 1/N
64	204	211	256	239
128	252	262	308	287
256	297	292	302	295
512	259	265	263	266
1024	257	258	259	261
2048	205	201	239	236

Future Plans for C67

- Release the Core Lite profile library for the C67 platform
- Explore releasing Core profile library
- Explore possibilities of partnering with OS vendors such as OSE Systems

P4-related Issues, I

- Single-precision optimization not a big concern outside embedded computing…
- Good free libraries exist (e.g., FFTW, LAPACK) and MKL exists as alternatives
- Academic basic kernels for matrix multiplication (non-ATLAS) are now mature enough to use with small code size, but these are not open source/redistributable (e.g., libgoto)
- Several universities working on better free libraries
- Code bloat an issue for certain library architectures when considering embedded (e.g., ATLAS code size)
- The merger of free libraries and free VSIPL has been tried, not as good a fully optimized library (e.g., VSIPL ERI, VSIPL Ref Implementation upgrade)
- Demand for commercial VSIPL for P4 remains a question

P4-related Issues, II

- Distinct flavors of P4 (e.g., Athlon) have distinctively different optimal libraries
  - Cache architecture
  - Instruction decode differences
  - TLB and other memory issues
  - Register file differences (e.g., 16 vs 8)
- Strong potential that future embedded P4 clones will also have different optimal choices in their hardware configurations

Why we think it is useful to have commercial VSIPL on P4 and C67

- Shows true performance portability story between diverse architectures, not just different G4/Altivec OS’s and vendors
- Allows system designers to work with assumption low software porting cost, and explore other aspects of design alternatives
- Processors are getting harder to program
- Precise mix of required optimizations for embedded not strong emphasis of free libraries per se

Abstract Presentation

Back to Agenda
Next Abstract

Conclusions

- Demand for VSIPL for non-G4 platforms is TBD… appears promising but not well developed
- Opportunities to achieve extremely high performance on clearly different architectures now evident
- Proof of concept may help drive adoption
- Technical hurdles involving hand-optimization remain for key inner kernels on each new platform, but do not require massive coding in assembly language if handled correctly
- C/C++ toolchain always an issue for new processor + OS combinations